David RENSHAW, *et al.*
Serial No.:09/452,927

--2--

<u>REMARKS</u>

Claims 1-15, 18-32 and 35 are currently pending in the application. Applicant requests reconsideration of the rejections in view of the following remarks.

35 U.S.C. §103 Rejection

Claims 1-4, 7, 18-21, 24 and 35 were rejected under 35 U.S.C. §103(a) for being unpatentable over U.S. Patent No. 5,999,729 issued to Tabloski, *et al.* ("Tabloski") in view of U.S. Patent No. 5,956,513 issued to McLain, *et al.* ("McLain") further in view of U.S. Patent No. 6,105,119 to Kerr, *et al.* ("Kerr"). Claims 5, 6, 8-13, 22, 23 and 25-30 were rejected under 35 U.S.C. § 103(a) over Tabloski in view of McLain further in view of Kerr and further in view of U.S. Patent No. 6,018,627 to Iyengar, *et al.* ("Iyengar"). Claims 14, 15, 31 and 32 were rejected under 35 U.S.C. § 103(a) over Tabloski in view of McLain further and in view of Kerr and U.S. Patent No. 6,083,276 to Davidson, *et al.* ("Davidson"). These rejections are respectfully traversed.

Applicants note that a §103 rejection requires the Examiner to first establish a prima facie case of obviousness: "The examiner bears the initial burden of factually supporting any prima facie conclusion of obviousness. If the examiner does not produce a prima facie case, the applicant is under no obligation to submit evidence of nonobviousness." M.P.E.P. § 2142. The Court of Appeals for the Federal Circuit has set forth three elements which must be shown for prima facie obviousness:

> To establish a prima facie case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both

be found in the prior art, and not based on applicant's disclosure.
In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991).

Because neither Tabloski, McLain or Kerr show running an executable to generate a data file containing definition files which are interpreted by a third-party computer system for future application execution, the Examiner has failed to establish a *prima facie* case of obvious over the pending claims.

The invention is directed to a form data files generator and relates to visual programming, and more particularly, to the generation of form data files which define electronic forms using such technology. In one aspect of the invention, a method for creating a data file using a programming development environment on a computer system including the step of building a program in the development environment to represent a data file is provided. The embodiment includes compiling the program in the development environment into a software executable. The executable generates the data file containing definition files which are interpreted by a third-party computer system for future application execution. Additionally, the third-party computer system comprises a dialog management system for a computer telephony system. As such, embodiments of the invention provide a method of using existing visual programming tools and environments to automatically generate textual definitions for forms thereby avoiding a need to make further large investments in next-generation programming tools.

Claims 1-4, 7, 18-21, 24 and 35

The Examiner asserts that Tabloski shows, in part, building a program to represent a data file, compiling it into a software executable, and running the executable to generate the data file. However, Tabloski actually shows a code generated by a program composition module where the code comprises a high-level programming language which is then compiled by a compiler to be executed by a parallel computer. Col. 20, lns. 20-35. Typically, the high-level language is C++. This is in contrast to claim 1, where an executable is run to generate a data file containing definition files which is then run by another computer. As such, the executable of claim 1

generates a data file to be run, while the executable program of Tabloski is itself executed by another computer.

In any event, the Examiner aknowledges that Tabloski does not explicitly disclose wherein the data file containing definition which are interpreted by a third-party computer system, and then cites McLain to show this feature. However, regardless of whether McLain supplies this missing feature of Tabloski, McLain fails to show running the executable to generate a data file and thus fails to supply the missing feature of Tabloski as discussed above. McLain actually shows a configuration data file ("CDF") which is a text file that is created by a user. Col. 7, lns. 48-49. As such, because the CDF is a text file that is created by a user, McLain fails to show running an executable to generate the data file. Thus, McLain fails to cure the deficiencies of Tabloski as described above.

The Examiner next cites Kerr for disclosing use of data files (header files) in third-party systems and vendors in a dialog system. But, Kerr fails to cure the deficiencies of Tabloski and McLain, as discussed above. In particular, rather than showing running an executable to generate a data file containing definition files which are interpreted by a third-party computer system for future application execution, Kerr is directed to data transfer circuitry having source code which is configured to lead to identically located data structures, no matter what type of compiler is used. Col. 21, lns. 41-42. Furthermore, Kerr shows using the common data structure to control data flow through system memory space so that the code is processor-independent. Col. 21, lns. 62-65. Accordingly, Kerr is directed to showing technology which may be applied at any point in a PC system and fails to disclose or suggest running an executable to generate a data file containing definition files which are interpreted by a third-party computer system for future application execution, and thus fails to cure the deficiencies of Tabloski and McLain as discussed above.

Accordingly, for the reasons discussed above, neither Tabloski, McLain or Kerr, either alone or in combination, disclose or suggest running an executable to generate a data file containing definition files which are interpreted by a third-party computer system for future

application execution, as set forth in claims 1, 18 and 35. As such, the Examiner has failed to establish a *prima facie* case of obviousness over claims 1, 18 and 35 and such claims are in allowable condition. Claims 2-15 and 19-32 are allowable at least for the reasons discussed above with respect to independent claims 1 and 18, from which they respectively depend, as well as for their added features. Applicant respectfully requests the rejection of claims 1-4, 7, 18-21, 24 and 35 be withdrawn.

Claims 5, 6, 8-13, 22, 23 and 25-30.

The Examiner asserts that Iyengar shows running an executable and at least one compiled component creates a file output stream and writes its respective data file to the output screen. However, Iyengar actually shows that each tool employed during a development process puts information into a repository and takes information out of the repository, and, as such, the system integrates tools used in different parts of the development process by passing necessary information from one tool to another. Accordingly, Iyengar fails to show that on running an executable at least one compiled component creates a file output stream and writes its respective data file information to the output stream, as set forth in claim 5.

More specifically, Iyengar is directed to a tool-independent system for application building in an object oriented development environment with data stored in a repository in OMG-compliant UML representation. Iyengar shows that output data from a developmental tool is transferred into a generic format data which is saved in a repository. The repository also contains all output data, application components, and information as to the relationship between the entities and objects stored in the repository. Each tool employed during the development process puts information into the repository and takes information out of the repository. In this way, the system integrates the tools used in different parts of the development process by passing necessary information from one tool to another. But, Iyengar fails to disclose or suggest running an executable to generate a data file containing definition files which are interpreted by a third-party computer system for future application execution, in contrast to the Examiner's assertion.

In fact, Iyengar does not discuss executable programs generating data files in any way because Iyengar is directed to a tool-independent system for application building. Additionally, the inputting and outputting of information to and from the repository is not done by running an executable file, and furthermore, such information is not interpreted by a third-party computer system for future application execution.

Accordingly, Iyengar fails to cure the deficiencies of Tabloski, McLain, and Kerr as discussed above, and claims 1, 18 and 35 are allowable over Tabloski, McLain, Kerr and Iyengar. Claims 5, 6, 8-13, 22, 23 and 25-30 are allowable at least for the reasons discussed above with respect to claims 1, 18 and 35, from which they respectively depend, as well as for their added features. Applicant respectfully requests the rejection of claims 5, 6, 8-13, 22, 23, and 25-30 be withdrawn.

Claims 14, 15, 31 and 32.

The Examiner cites Davidson for showing disclosing data files comprising mark-up information. As such, Davidson fails to cure the deficiencies of Tabloski, McLain and Kerr as discussed above with respect to independent claims 1, 18 and 35. Accordingly, claims 1, 18 and 35 are allowable over the combination of Tabloski, McLain, Kerr and Davidson. Claims 14, 15, 31 and 32 are allowable at least for the reasons discussed above with respect to claims 1, 18 and 35, from which they respectively depend as well, as for their added features. Applicant respectfully requests that the rejection of claims 14, 15, 31 and 32 be withdrawn.
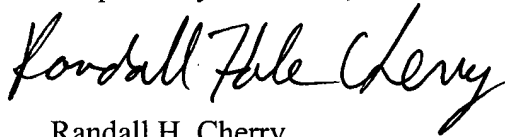
CONCLUSION

In view of the foregoing amendments and remarks, Applicant submits that all of the claims are patentably distinct from the prior art of record and are in condition for allowance. The Examiner is respectfully requested to pass the above application to issue. The Examiner is invited to contact the undersigned at the telephone number listed below, if needed. Applicant hereby makes a written conditional petition for extension of time, if required. Please charge any deficiencies in fees and credit any overpayment of fees to Attorney's Deposit Account No. 09-0457.

Respectfully submitted,

Randall H. Cherry
Registration No. 51,556

Andrew M. Calderon
Registration No. 38,093

McGuireWoods, LLP
Suite 1800
1750 Tysons Blvd.
McLean, VA 22102
(703) 712-5426

\\COM\406336.2